

Optimal Sensor Positioning (OSP); A Probability Perspective Study

Sung Ha Kang, Seong Jun Kim and Haomin Zhou*

Abstract

We propose a method to optimally position a sensor system, which consists of multiple sensors, each has limited range and viewing angle, and they may fail with a certain failure rate. The goal is to find the optimal locations as well as the viewing directions of all the sensors and achieve the maximal surveillance of the known environment.

We setup the problem using the level set framework. Both the environment and the viewing range of the sensors are represented by level set functions. Then we solve a system of ordinary differential equations (ODEs) to find the optimal viewing directions and locations of all sensors together. Furthermore, we use the intermittent diffusion, which converts the ODEs into stochastic differential equations (SDEs), to find the global maximum of the total surveillance area. The numerical examples include various failure rates of sensors, different rate of importance of surveillance region, and 3-D setups. They show the effectiveness of the proposed method.

1 Introduction

We propose a computational method to position a surveillance system with multiple sensors, where each sensor has a limited viewing angle and range. Some of the sensors may have a non-zero failure rate. Our goal is to achieve the maximum surveillance of a complicated environment.

The increasing societal demands for security lead to a growing need for surveillance activities in various environments, such as public places like transportation infrastructures and parking lots, shopping malls and telecommunication relay towers. How to optimally position the sensors is an interesting, important and maybe challenging problem, which can directly impact the public safety and security. As reported in [17], the optimal position of sensors may also provide better resources allocation and system performance.

In practice, sensors used in many applications often come with limited sensing ability, for example, an object cannot be detected if its distance to the sensor is too far (larger than $r > 0$) or it is located outside of the sensors viewing angle ($[v, v + \theta]$, $0 < \theta < 2\pi$, where v is a viewing direction and θ is a limited width of viewing angle). There may exist obstacles that block the view of the sensors. In addition, with certain probability, sensors may fail.

There exist many related studies in optimal sensor positioning. Using graph-based approaches, this problem can be posed as a combinatorial optimization problem which has been shown to be NP-hard [13]. Thus, simple enumeration and search techniques, as well as general purpose algorithms may experience extraordinary difficulty in some cases. The placements of sensors have

*School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, USA email: kang@math.gatech.edu; skim396@math.gatech.edu; hmzhou@math.gatech.edu

This work is partially supported by NSF Awards DMS-1042998, DMS-1419027, ONR Award N000141310408, and Simons Foundation grant 282311.

been undertaken for the well-known *art gallery* problem, a classical problem that aims at placing stationary observers to maximize the surveillance area [1] in a cluttered region. The art gallery problem and its variations have drawn considerable attention in recent decades, for example in sensor placements [4, 8, 9, 19, 25]. Similar studies have been conducted to other problems such as routes for patrol guards with limited visibility or limited mobility. We refer readers to a book [18] and papers [11, 12, 20, 23] for more references therein. Problems with possible sensor failures have also been studied. For example, the use of a Bayesian approach is discussed in [2], incorporating environmental data, such as temperatures, from multiple sensors to gather spatial statistics can be found in [5]. A Gaussian process model is widely used to predict the effect of placing sensors at particular locations [6, 13, 24, 27, 28]. See also [10] for optimally placing unreliable sensors in a 1-dimensional environment.

Despite of the existence of extensive literature on related topics, the problem that we are interested in remains to be challenging. The main difficulties often come from following aspects: (1) The obstacles have arbitrary shapes. (2) Finding a feasible placement with largest coverage area is often costly because it is an infinite dimensional problem. (3) Finding the globally optimal position, if possible, is usually computationally intractable. In addition, we consider the case of each sensor having a certain failure rate.

In this paper, we tackle the problem using the level set formulation together with the intermittent diffusion, a stochastic differential equation (SDE) based global optimization strategy [26]. The level set framework gives us the freedom to consider various shapes of obstacles and a way to incorporate the limited coverage into the definition of the coverage level set function. The intermittent diffusion is then used to find one of the optimal viewing directions and locations in all of the feasible positions which gives the largest surveillance area of the environment.

The layout of the paper is as follows. In Section 2, the level set formulation of limited sensing range and viewing angle is introduced. We propose methods to find optimal locations and viewing directions in this setting. To have a possibility of finding a global optimum, we use the intermittent diffusion. In Section 3, we describe the details of having a nonzero failure rate for the sensors and optimal condition for surveilling an area. Numerical implementation details and different applications are given in Section 4. We conclude by giving a summary and discussion in Section 5.

2 Optimal positioning of the sensors

In this section, we introduce the level set formulation for the coverage function of sensors with limited sensing range and limited width of viewing angle. Then we propose a strategy to find the optimal viewing directions of multiple stationary sensors to achieve the largest surveillance of the environment. Finally, we introduce the intermittent diffusion to find the globally optimal placement.

2.1 Level set formulation for the sensors coverage

In the level set framework, the environment is described by a level set function, say $\psi(x)$, with positive values outside of the obstacles, negative values inside, and the zero level curves representing boundaries of obstacles.

A related application on visibility problem uses this level set setting [3, 21], where light rays from a vantage point travel in straight lines and are obliterated upon contact with the surface of an obstacle. A point is seen by the observer (or sensor) placed at a given vantage point, if the line segment between the point and the vantage point does not intersect any of the obstructions. In existing work [12, 14, 15, 16] using such a setting, the observer has either a circular coverage

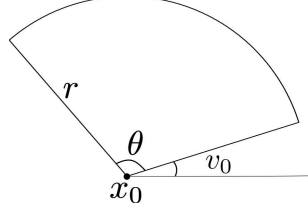


Figure 2.1: A sensor with a finite sensing range and a limited width viewing angle. The coverage area of the sensor is described by a circular sector. x is the location of the sensor, r is the limited sensing range, θ is the limited width of viewing angle, and v is the viewing direction.

(viewed) region, or an infinite range. In this paper, we further consider the case of limited width of viewing angle with a finite range of coverage region.

We denote the computational domain as D , a compact subset of \mathbb{R}^2 . Let Ω be the collection of obstacles, i.e. a closed set containing finite number of connected components comprising one or multiple given obstacles in D . The level set function $\psi(x)$ represents the environment: inside the obstacles Ω is negative, and outside is positive. In our algorithms, we define $\psi(x)$ as the typical signed distance function from the boundaries of Ω . A sensor located at a point x in $D \setminus \Omega$, has the line-of-sights as straight line segments originated from x and ended at y with either $|x - y| = r$ and $y \in D \setminus \Omega$, or $|x - y| \leq r$ and y is a point on the boundary of the obstacles. The coverage region of the sensor is the union of all line-of-sights emanating from the sensor and the angle to the horizontal axis is in $[v, v + \theta]$. Figure 2.1 depicts the coverage area of the sensor in the setup.

We use a normalized viewing vector $\vec{n}_x(y) = \frac{y-x}{|y-x|}$ which connects an initial point x with a terminal point y . Then, we define the sensor coverage as a level set function ϕ given by

$$\phi(y; x, r, v) = \min_{z \in \mathcal{L}(x, y)} \{ \psi(z), r - |z - x|, \varphi(x, y) \} \quad (2.1)$$

where $\mathcal{L}(x, y)$ is the line segment connecting x and y , and $\varphi(x, y) = 1$ if the angle of $\vec{n}_x(y)$ from the horizontal axis lies in $[v, v + \theta]$ and $\varphi(x, y) = -1$, otherwise. This way the function $\phi(\cdot; x, r)$ represents the coverage in a bounded domain D by

$$\begin{cases} \phi(\cdot; x, r, v) > 0, & \text{covered regions,} \\ \phi(\cdot; x, r, v) \leq 0, & \text{non-covered regions.} \end{cases} \quad (2.2)$$

With the coverage level set function $\phi(y; x, r, v)$ we can calculate the coverage area of the sensor at x with the viewing direction v by

$$V(x, v) = \int_D H(\phi(y; x, r, v)) dy$$

where H denotes the one-dimensional Heaviside function.

To compute the coverage function $\phi(y; x, r, v)$, we present a simple strategy in Algorithm 1 which follows the PDE based strategy presented in [21]. The main idea is to use the method of characteristics for the nonlinear first order PDEs, and we modify to handle the finite sensing range and the limited width of viewing angle.

We solve (2.3) using the upwind finite differencing scheme. Note the computational workload in each iteration is dominated by the second step. However, these steps are only needed for computing ϕ inside the sensor's coverage range, so we can save the computations.

Algorithm 1 PDE based algorithm to compute coverage

Input: level set function ψ , a sensor $x \in D \setminus \Omega$, radius r , and angles v, θ .

1. Set $\phi(y) = \psi(y)$ for $y \in D$.

2. If $y \in B_r(x)$, solve for ϕ :

$$\nabla_y \phi(y; x, r, v) \cdot \vec{v}_x(y) = 0, \quad (2.3)$$

subject to the boundary condition $\phi = \psi$.

If else, set $\phi(y; x, r, v) = -1$.

3. Update $\phi = \min(\phi, \psi, r - |y - x|, \varphi(x, y))$.

Generalization of the above setting for multiple sensors is immediate. Let $\{x_1, \dots, x_m\}$ denote the location of m sensors. We also let $\{r_1, \dots, r_m\}$ be the sensing range of the individual sensors, $\{\theta_1, \dots, \theta_m\}$ be the limited width of viewing angle, and $\{v_1, \dots, v_m\}$ be the viewing direction of each, respectively. The coverage of multiple sensors is the union of the coverage of all sensors. Similar to (2.1), we define the coverage level set function with respect to multiple sensors by

$$\phi(y; x_1, \dots, x_m, r_1, \dots, r_m, v_1, \dots, v_m) = \max_{i=1, \dots, m} \phi(y; x_i, r_i, v_i)$$

where the associated $\varphi(x_i, y)$ in ϕ is 1 if the angle of $\vec{n}_{x_i}(y)$ from the horizontal axis is in $[v_i, v_i + \theta_i]$ and $\varphi(x_i, y) = -1$, otherwise. We apply Algorithm 1 to compute $\phi(y; x_i, r_i, v_i)$ with respect to each sensor, and then take the union to obtain the combined coverage. Moreover, the coverage area of multiple sensors with respect to the coverage range $\{r_1, \dots, r_m\}$ and viewing angles $\{v_1, \dots, v_m\}$ is given by

$$V(x_1, \dots, x_m, v_1, \dots, v_m) = \int_{\Omega} H(\phi(y; x_1, \dots, x_m, r_1, \dots, r_m, v_1, \dots, v_m)) dy. \quad (2.4)$$

We note that V is Lipschitz continuous under the condition that there is no fattening in the level set function ϕ near the boundary of the obstacles. The details can be found in [3].

In the following sections, we focus on maximizing this coverage area with multiple sensors.

2.2 Optimal viewing direction (fixed locations)

We propose a procedure for the optimal position of multiple stationary sensors to maximize the surveillance area. Our approach incorporates the previous level set formulation into the optimization methods. We first consider the case when the location of sensors is fixed and only the viewing direction is allowed to change. The goal is to find the optimal viewing direction of each sensor that maximizes the coverage area V in (2.4) with fixed locations $\{x_1, \dots, x_m\}$. We use the gradient ascent method, i.e., following the gradient flow to steer the viewing direction:

$$\partial_t v_i = \nabla_{v_i} V(v_1, \dots, v_m), \quad i = 1, 2, \dots, m, \quad (2.5)$$

where ∇_{v_i} is the gradient operator with respect to the angle v_i .

To solve (2.5) numerically, we use the forward difference to approximate the time derivative and the central difference to approximate the gradient operator and, which leads to

$$D_+^k v_i = D_0^{h_v} V(v_1, \dots, v_m), \quad i = 1, 2, \dots, m, \quad (2.6)$$

where k is a temporal step size and h_v is the step size for the angle. The gradient of V requires to compute $\phi(y; x_i, r_i, v_i \pm h_v)$. We use the Algorithm 1 for computing ϕ over the grid with equally

spaced points with a spacing of spatial step size h in the computational domain D . In the evaluation of V , we numerically integrate $H(\phi)$ using a trapezoidal rule over D and regularize the Heaviside function by the method proposed in [7]:

$$H_\epsilon(\phi) = \begin{cases} 1, & \phi \geq \epsilon, \\ \frac{1}{2}(1 + \frac{\phi}{\epsilon}), & |\phi| < \epsilon, \\ 0, & \phi \leq -\epsilon, \end{cases}$$

using the point-wise scaling with l_1 norm of $\nabla\phi$, $\epsilon = \frac{h}{2}|\nabla\phi|_{l_1}$.

The gradient method (2.6) finds a local rather than a global optimum. To find the globally optimal solution, we employ the intermittent diffusion method [26] which adds random perturbations to the gradient flow (2.6). This helps the deployment move out of local traps and have a chance to find the globally optimal one. The following SDE is proposed to be solved before the gradient ascent flow,

$$dv_i(t) = \nabla_{v_i} V(v_1, \dots, v_m)dt + \sigma(t)dW(t), \quad i = 1, \dots, m, \quad (2.7)$$

where $W(t)$ is the Brownian motion in \mathbb{R}^m , and $\sigma(t)$ is a piecewise constant function alternating between zero and a random positive constant. The random component perturbs the initial condition for the gradient ascent so that the solution can escape from the trap of a local maximizer and approach other ones.

Algorithm 2 Optimal viewing direction for globally maximal coverage

Input: sensors $\{x_1, \dots, x_m\}$, radii of coverage $\{r_1, \dots, r_m\}$, viewing directions $\{v_1, \dots, v_m\}$, limited widths $\{\theta_1, \dots, \theta_m\}$, step size for location h_x , step size for angle h_v , and temporal step size k .

Initialize: Randomly locate the sensors on the allowable area.

Repeat: Set $j = 1$ and iterate N times to obtain N different sets of positioning.

1. Set α as the scale for diffusion strength, and γ the scale for diffusion time. Let $\sigma := \alpha d$, and $T := \gamma t$ where two positive random numbers d, t within $[0, 1]$ by uniform distribution.
2. Set the optimal deployment $L_{opt} = (v_1, \dots, v_m)$.
3. Taking L_{opt} as the initial condition, compute the SDE for $t \in [0, T]$, $i = 1, \dots, m$.

$$dv_i(t) = \nabla_{v_i} V(v_1, \dots, v_m)dt + \sigma(t)dW(t), \quad i = 1, \dots, m,$$

and record the final configuration as $L_{ini} = (v_1, \dots, v_m)$.

4. Compute the gradient ascent flow until convergence with the initial condition L_{ini}

$$\partial_t v_i = \nabla_{v_i} V(v_1, \dots, v_m), \quad i = 1, 2, \dots, m$$

and record the final configuration as $L_{candidate}$.

5. If $V(L_{candidate}) > V(L_{opt})$, set $L_{opt} = L_{candidate}$.
 6. Repeat with $j = j + 1$.
-

One iteration of the procedure is to solve a SDE (2.7) for $t \in [0, T]$ and then compute the gradient ascent flow (2.6) until convergence. This procedure finds one of the globally optimal solutions with probability arbitrarily close to one. Indeed, denoting $\alpha \in (0, 1)$ by the probability that one iteration attains a close approximation to the globally optimal solution, the rate of convergence is proven to be $1 - (1 - \alpha)^N$ with N iterations. See [26] for further details.

Figure 2.2 illustrates a result of the algorithm applied to 8 stationary sensors to cover the environment with polygonal obstacles. For simplicity, we assumed that all sensors have the same sensing range r and the width of viewing angle θ . In subfigure (a), the obstacles are the shaded three polygons, and the locations of each sensor are fixed at red crosses. The dotted sector represents the sensor's coverage if any obstruction is not given by an obstacle. Otherwise, the red (shaded) area inside the sector shows the coverage. The objective is now translated into finding the viewing directions such that the union of each coverage is largest. Algorithm 2 seeks one of the desired sets of the viewing direction $\{v_1, \dots, v_m\}$ which globally maximizes (2.4). The final result is presented in subfigure (b) with $N = 80$ iterations.

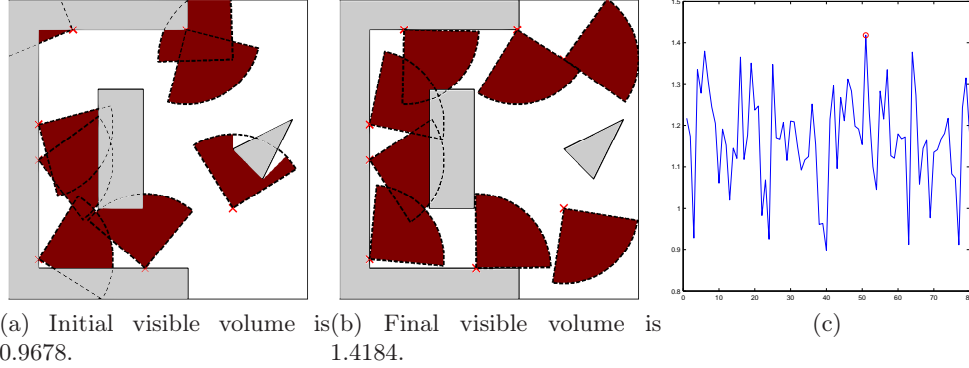


Figure 2.2: Optimal viewing directions for multiple sensors with fixed locations. The viewing directions of multiple sensors change from the image (a) to the image (b) by Algorithm 2. (c) The value of coverage area with respect to the iteration. The maximum coverage is achieved at the 51th iteration. Parameters are chosen as $r = 0.5$, $h = 0.005$, $\theta = \pi/2$, $h_v = 4h$, and $h_x = h$.

2.3 Optimal Sensor Positioning (OSP)

We now consider the case where one can adjust the location of sensors. The objective is extended to find optimal locations as well as viewing directions that globally maximize the coverage area V in (2.4). The optimal positioning problem can be reformulated as a $2m$ -dimensional problem:

$$\max_{x_i, v_i \in \mathcal{A}} V(x_1, \dots, x_m, v_1, \dots, v_m)$$

where \mathcal{A} is the feasible set of sensors' locations and viewing directions. Typical locations of the sensors are on the boundary of obstacles $\partial\Omega$.

Among the sensors, we denote a subset of sensors whose locations are adjustable by $\{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{\tilde{m}}\}$, i.e., we take $\{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{\tilde{m}}\} \subseteq \{x_1, \dots, x_m\}$, and each $\tilde{x}_{\tilde{i}}$ is uniquely identified with some sensor x_i . Extending the previous approach for fixed sensors, we employ the intermittent diffusion and the gradient ascent to maximize the coverage area. However, due to the adjustability of some sensors, a parametrization of the boundary of obstacles is required to conduct the optimization with respect to the location. This can be done by using a reinitialization method given in [22] and parameterizing zero level curves which represent boundaries of obstacles.

Now, the gradient ascent is used for the modified equation of (2.5):

$$\begin{aligned} \partial_t v_i &= \nabla_{v_i} V(x_1, \dots, x_m, v_1, \dots, v_m), \quad i = 1, 2, \dots, m, \\ \partial_t \tilde{x}_j &= \nabla_{\tilde{x}_j} V(x_1, \dots, x_m, v_1, \dots, v_m), \quad j = 1, 2, \dots, \tilde{m}, \end{aligned} \quad (2.8)$$

where ∇_{v_i} is the gradient operator with respect to the viewing direction v_i , and $\nabla_{\tilde{x}_j}$ is the gradient operator with respect to the location \tilde{x}_j along the boundary of obstacle where the sensor \tilde{x}_j lies in.

As in (2.5), when the solution reaches a steady state, this gives one of many local optimal solutions. To find a globally optimal solution, we use the intermittent diffusion. It adds random perturbations intermittently to the gradient flow (2.8) which leads to the following SDEs,

$$\begin{aligned}\partial_t v_i &= \nabla_{v_i} V(x_1, \dots, x_m, v_1, \dots, v_m) dt + \sigma_1(t) dW_1(t), \quad i = 1, 2, \dots, m, \\ \partial_t \tilde{x}_j &= \nabla_{\tilde{x}_j} V(x_1, \dots, x_m, v_1, \dots, v_m) dt + \sigma_2(t) dW_2(t), \quad j = 1, 2, \dots, \tilde{m},\end{aligned}\quad (2.9)$$

where $W_1(t)$ and $W_2(t)$ are the Brownian motions in \mathbb{R}^m and $\mathbb{R}^{\tilde{m}}$, respectively. In addition, $\sigma_1(t)$ and $\sigma_2(t)$ are piecewise constant functions alternating between zero and a random positive constant. This set of SDEs is followed by the gradient ascent which can escape from the trap of a local minimizer and approach other ones.

Algorithm 3 presents Position Optimization for Sensors (POS) for maximal coverage area. Here we use the term positioning to describe both locations and viewing directions.

Algorithm 3 OSP

Input: sensors $\{x_1, \dots, x_m\}$, radii of coverage $\{r_1, \dots, r_m\}$, viewing directions $\{v_1, \dots, v_m\}$, limited widths $\{\theta_1, \dots, \theta_m\}$, step size for location h_x , step size for angle h_v , and temporal step size k . Also, adjustable sensors $\{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{\tilde{m}}\} \subseteq \{x_1, \dots, x_m\}$.

Initialize: Randomly locate the sensors on the allowable area.

Repeat: Set $j = 1$ and iterate N times to obtain N different deployments.

1. Set α as the scale for diffusion strength, and γ the scale for diffusion time. Let $\sigma := \alpha d$, and $T := \gamma t$ where two positive random numbers d, t within $[0, 1]$ by uniform distribution.
 2. Set the optimal deployment $L_{opt} = (x_1, \dots, x_m, v_1, \dots, v_m)$.
 3. Taking L_{opt} as the initial condition, compute the SDE (2.9) for $t \in [0, T]$ and record the final configuration as $L_{ini} = (x_1, \dots, x_m, v_1, \dots, v_m)$.
 4. Compute the gradient ascent flow (2.8) until convergence with the initial condition L_{ini} and record the final configuration as $L_{candidate}$.
 5. If $V(L_{candidate}) > V(L_{opt})$, set $L_{opt} = L_{candidate}$.
 6. Repeat with $j = j + 1$.
-

Evaluations of $\nabla_0^{h_v} V$ and $\nabla_0^{h_x} V$ require to compute $\phi(y; x_i, r_i, v_i \pm h_v)$ for $i = 1, \dots, m$ and $\phi(y; \tilde{x}_j \pm h_x, r_i, v_i)$ for $i = 1, \dots, \tilde{m}$ if x_j is allowed to move along the boundary. In Step 4, we use Euler's method to update v_i by $v_i + k D_0^{h_v} V(x_1, \dots, x_m, v_1, \dots, v_i, \dots, v_m)$ for $i = 1, \dots, m$ and \tilde{x}_j by $\tilde{x}_j + k D_0^{h_x} V(x_1, \dots, \tilde{x}_j, \dots, x_m, v_1, \dots, v_m)$ for movable \tilde{x}_j , respectively. Figure 2.3 describes a result of Algorithm 3 applied to the same environment in Figure 2.2, but all of the sensors are movable except for the two sensors not on the boundary of any obstacle. An optimal position which locally maximizes (2.4) is presented in subfigure (b). It is evident that the resulting coverage area is much larger than the one in Figure 2.2 because mobile sensors have more degrees of freedom than that of stationary sensors. For simplicity, we suppose that all sensors have the same sensing range r and width of viewing angle θ .

3 Optimal sensor positioning with a failure rate

The failure of sensors is crucial for the surveillance system to achieve its intended task, and this can be added to the stage of sensor positioning. We extend the previous approach to find optimal

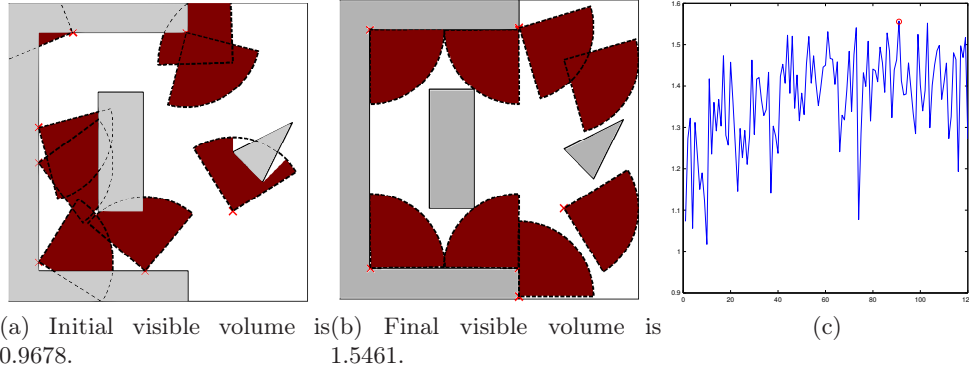


Figure 2.3: Optimal locations and viewing directions of multiple sensors. The locations and viewing directions of multiple sensors change from the image (a) to the image (b) by Algorithm 3. (c) The value of coverage area with respect to the iteration. The maximum coverage is achieved at the 91st iteration. Parameters are chosen as $r = 0.5$, $h = 0.005$, $\theta = \pi/2$, $h_v = 4h$, and $h_x = h$.

positions when the sensors have a non-zero probability of failure. As a priori estimate, we consider an expected coverage area of the sensors in the objective functional and analyze an optimal solution to maximize it. If there are sufficiently many sensors in the sense that the collective coverage area is much larger than the area of $D \setminus \Omega$, the optimal solution will be shown to uniformly spread the sensors over the monitoring area.

We assume that sensors fail independently, i.e., if two sensors have a failure probability p_1 and p_2 , respectively, then $p_1 p_2$ is the joint failure probability. Given a point y in the environment, we denote $0 < p(y) < 1$ by a probability that a set of sensors fails to detect y . If y is not in any of the sensor's coverage, then $p(y)$ is set to be 1. Furthermore, we denote $p_k(y)$ by a probability that the individual k -th sensor fails to detect a point y .

The expectation of the coverage area with possible sensor failures is defined as

$$\mathbb{E}(V(x_1, \dots, x_m, v_1, \dots, v_m)) = \int_D H(\phi(y; x_1, \dots, x_m, r_1, \dots, r_m, v_1, \dots, v_m))(1 - p(y)) dy. \quad (3.1)$$

That is, the functional is the integral of the products of covered area with the probabilities that the sensors to cover it are working. Note that if there is no sensor failure, then $p(y) = 0$ so that the objective functional (3.1) is reduced to (2.4). The optimization problem is to maximize the expected area of coverage which is formulated as:

$$\max_{x_i, v_i} \mathbb{E}(V(x_1, \dots, x_m, v_1, \dots, v_m)) \quad (3.2)$$

subject to the locations of the sensors adjustable on either the boundary of obstacles $\partial\Omega$ or the boundary of the computational domain.

For simplicity, we discretize a two dimensional environment by uniformly placing a grid (ih, jh) on it and denote $0 < p_k(i, j) < 1$ by a probability that the k -th sensor fails to detect a grid point (ih, jh) . With an uniformly space grid, the expected area of sensors' coverage is numerically approximated by

$$\sum_{i,j} h^2 \left(1 - \prod_{k=1}^n p_k(i, j) \right).$$

The following considers how the multiple sensors should be positioned in order to maximize the objective functional (3.1). We show that positioning multiple sensors with less overlapping area is closer to the optimal solution than doing so with more overlapping area.

Proposition 1. *Positioning multiple sensors with minimal overlapping regions attains the greater expected coverage area than those with more overlapping regions.*

Proof. Suppose that the small region \mathcal{R} with an area B is given, and that this region is fully covered by m sensors. Firstly, suppose that their coverages do not intersect. We denote $\mathcal{R}_{nonoverlap}$ by the sensor positions where all sensors do not intersect and cover entire \mathcal{R} . Then the contribution of this sensor positioning to the total expected coverage area is

$$\mathbb{E}[\mathcal{R}_{nonoverlap}] = B \left(\sum_{k=1}^m (1 - p_k) \right) \quad (3.3)$$

where the k -th sensor has a failure probability p_k .

Secondly, suppose that several \tilde{m} ($\leq m$) sensors overlap and fully cover \mathcal{R} . In this case, the contribution to the total expected coverage area is

$$\mathbb{E}[\mathcal{R}_{overlap}] = B \left(1 - \prod_{k=1}^{\tilde{m}} p_k \right). \quad (3.4)$$

Now, we compare $\sum_{k=1}^m (1 - p_k)$ with $1 - \prod_{k=1}^{\tilde{m}} p_k$. Since the failure probability is between 0 and 1,

$$1 - \prod_{k=1}^m p_k > 1 - \prod_{k=1}^{\tilde{m}} p_k.$$

Using the mathematical induction, suppose that

$$\sum_{k=1}^m (1 - p_k) \geq 1 - \prod_{k=1}^m p_k$$

holds true. Then, multiplying p_{n+1} yields that

$$p_{n+1} \sum_{k=1}^n (1 - p_k) \geq p_{n+1} - \prod_{k=1}^{n+1} p_k.$$

Now,

$$\begin{aligned} \sum_{k=1}^{n+1} (1 - p_k) &\geq \sum_{k=1}^{n+1} (1 - p_k) - (1 - p_{n+1}) \sum_{k=1}^n (1 - p_k) \\ &= 1 - p_{n+1} + p_{n+1} \sum_{k=1}^n (1 - p_k) \\ &\geq 1 - \prod_{k=1}^{n+1} p_k. \end{aligned}$$

Therefore, $\mathbb{E}[\mathcal{R}_{nonoverlap}] \geq \mathbb{E}[\mathcal{R}_{overlap}]$. □

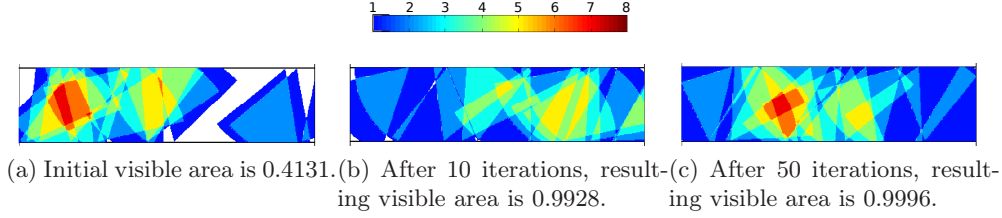


Figure 4.1: Position optimization with no possible failure. The resulting sensor positions are obtained using the parameters $r = 0.6$, $h = 0.005$, $\theta = \pi/3$. $N = 50$ iterations of our algorithm are used. The horizontal colorbar displays how many sensors cover the area using the colormap.

In other words, when the sensor fails with non-zero probability, spreading multiple sensors over the monitoring area as much as possible attains the largest expectation. Consequently, if the algorithm maximizes (3.1), the optimal position will not only cover all of the free areas, $D \setminus \Omega$, but also minimize the overlaps among the sensors. We compare two scenarios in the following section; optimal positioning of the sensors with or without a failure rate and test the performance of our proposed algorithm.

4 Numerical applications

In this section, various numerical experiments are given. We use Algorithm 3 with the functional (3.1) for most of the examples, considering a nonzero failure rate. In one example depicted in Figure 4.1 where the failure rate $p = 0$, we use the objective functional (2.4).

4.1 Comparison: failure rate $p = 0$ and $p \neq 0$

Two examples compare the sensor positioning with or without a failure rate. We first consider a problem with sensors which work perfectly, i.e., the probability of failure is $p = 0$. With a set of these sensors, a rectangular environment without obstacles will be monitored. Suppose that the number of sensors is large so that the environment can be fully covered with some redundancies. Figure 4.1(a) shows an initial position of 16 sensors which is randomly determined by putting 8 sensors for each side. We display the number of overlaps in the sensors coverage by color scales using a horizontal color bar. The area of the region to be covered is 1. With the finite sensing range $r = 0.6$ and the limited width of sensing angle $\theta = \pi/3$, the collective sensors' coverage area, if they are disjoint, is 3.016.

As shown in subfigures (b) and (c), with iterations our algorithm focuses on covering all of the regions without any empty space to achieve the optimal solution. The number of coverage overlaps is not considered.

In real applications, sensors may fail. Assuming that the sensor fails to work with a probability p , our algorithm maximizing (3.1) not only focuses on covering all of the free areas but also minimizes the overlaps among the sensors as proven in Proposition 1. We take the same environment of Figure 4.1, but assume that each sensor has a 50% chance of failure, i.e., with a probability $p = 0.5$. Figure 4.2 shows the results of our algorithm. Since we compute the expected area of covered region with a probability, the initial value of the objective functional is different from the one in Figure 4.1. The algorithm automatically finds the optimal deployment of sensors which leads to the full coverage as well as the minimal area of the intersections. The subfigure (c) shows evenly distributed coverages of sensors compared to Figure 4.1(c).

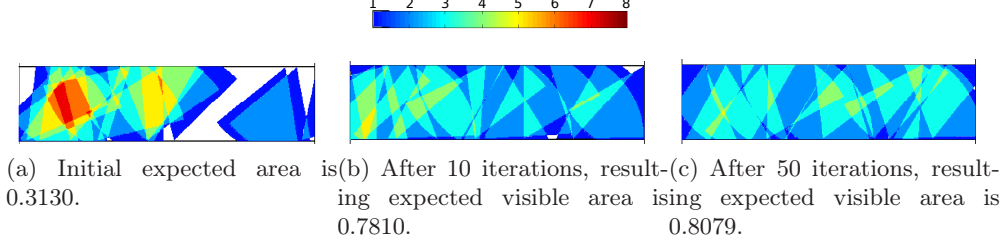


Figure 4.2: Position optimization with a possible failure. The resulting sensor placements are obtained using the parameters $r = 0.6$, $h = 0.005$, $\theta = \pi/3$. $N = 50$ iterations of our algorithm are used. Each sensor has a 50% chance of not working, $p = 0.5$. Compared to Figure 4.1, the result shows more uniformly distributed coverages.

4.2 General examples for $p \neq 0$

We consider more general shapes of environment. The first is a narrow alley with a corner. Given a sufficient number of sensors with the same failure rate, we will optimally position them on the walls to maximize an expectation of the collective coverage. The second is an arbitrary polygon, and we consider the problem of optimal positioning on the boundary.

Figure 4.3 illustrates how multiple sensors are optimally positioned to monitor the alley with a corner. 4 sensors are randomly located on each wall facing the alley, i.e., a total of 16 sensors, as in subfigure (a). Red crosses indicate the sensor locations, and sectors with dotted lines describe the sensors' coverage. The optimal positions found by our algorithm with 50 iterations are presented in subfigure (b). The coverage of sensors is uniformly distributed across the target area. This environment will be generalized to 3D in Section 4.5.

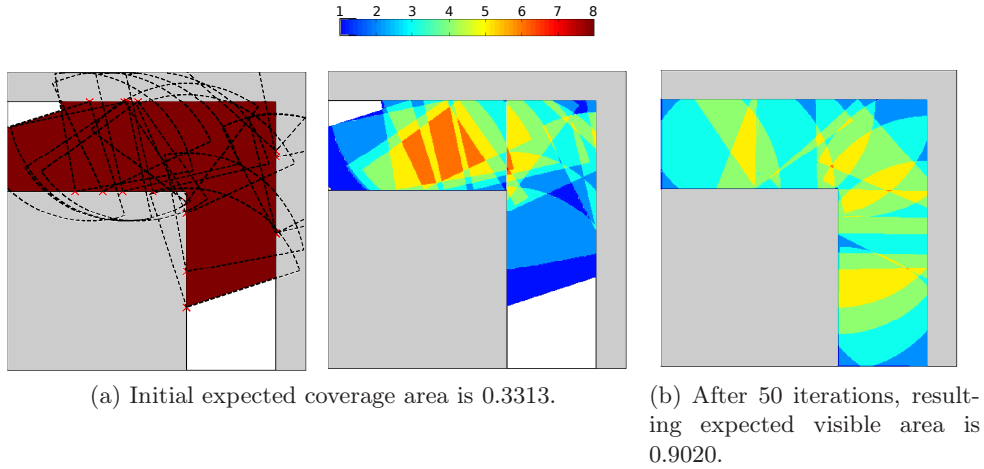


Figure 4.3: Optimal positioning in an alley with a corner. The resulting sensor positions are obtained using the parameters $r = 0.8$, $h = 0.005$, $\theta = \pi/2$, $p = 0.5$. 16 sensors are considered. The result shows uniformly distributed coverages.

In Figure 4.4, optimal sensor positioning in the environment with polygonal obstacles is considered. This scenario can be interpreted as positioning security surveillance cameras on the outside (e.g., rooftop) of urban buildings. The polygons are shaded gray, and the failure rate of the sensors is taken as 0.5. We apply our algorithm to optimally position 15 sensors on the boundary. The

result in subfigure (b) is obtained with 200 iterations. As shown in the figure, our algorithm can achieve a maximum area of coverage while handling complicated shapes of obstacles.

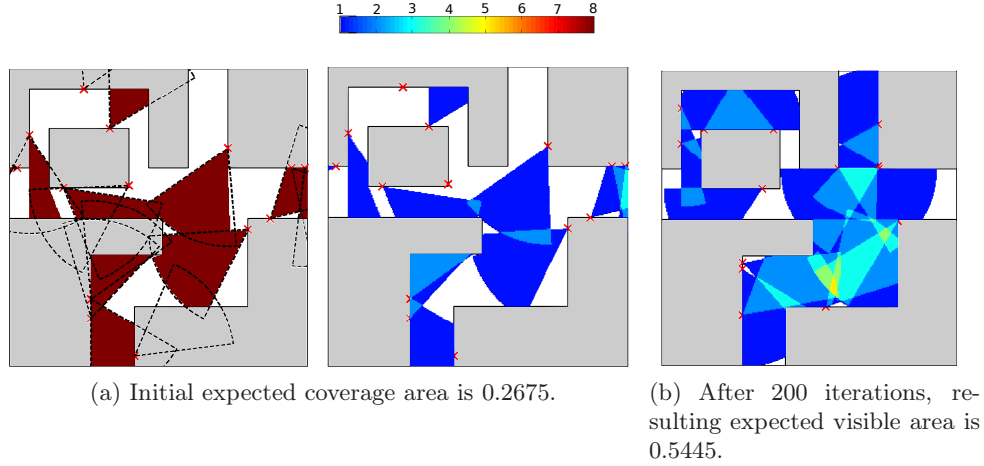


Figure 4.4: Optimal positioning in an environment with polygonal obstacles. The resulting sensor placements are obtained using the parameters $r = 0.5$, $h = 0.005$, $\theta = \pi/3$, $p = 0.5$. 15 sensors are considered. This algorithm handles a complicated environment when sensors can move along the boundary.

4.3 Access monitoring

Access monitoring system plays an important role in securing a building and its surroundings. Well-designed monitoring systems are able to detect all of the suspicious activities while considering a possible failure of the sensors. In this section, we consider a problem of optimally positioning the sensors to monitor the population entering a building.

We position the sensors on the boundary of the important building. The building is considered as the obstacle, and the sensors move along the boundary of the building to find the optimal positions. To use our approach with a possible failure rate, we construct the functional as

$$\mathbb{E}(V(x_1, \dots, x_m, v_1, \dots, v_m)) = \int_D \chi_A(y) H(\phi(y; x_1, \dots, x_m, r_1, \dots, r_m, v_1, \dots, v_m)) (1 - p(y)) dy$$

where χ_A is the indicator function of a set A , and $0 < p(y) < 1$ is a probability that a set of sensors fails to detect y . Here, the set A is a closed region which encloses the building so that the floating population must traverse this region in order to either enter or exit the building. Various shapes of A can be considered, and we use a narrow band around the building for simplicity. The main difference from the previous examples is that the area of full coverage only needs to be considered on a narrow band around the building as long as the access is fully covered.

In the following experiments, we consider a pentagon shaped building located at the center of the computational domain. Two red pentagons enclosing the building are the monitored region A . Note that one cannot move out or in to the building without passing the strip A . We employ the algorithm which maximizes the above functional to find the optimal position as well as the viewing direction of sensors while considering the sensor's possible failure.

In the case of symmetric objects, one can consider solving the problem more efficiently by adding a symmetry constraint to the feasible position. Figures 4.5(a-b) show such an example where we position three sensors to each side symmetrically. One sensor is fixed at the center with a viewing

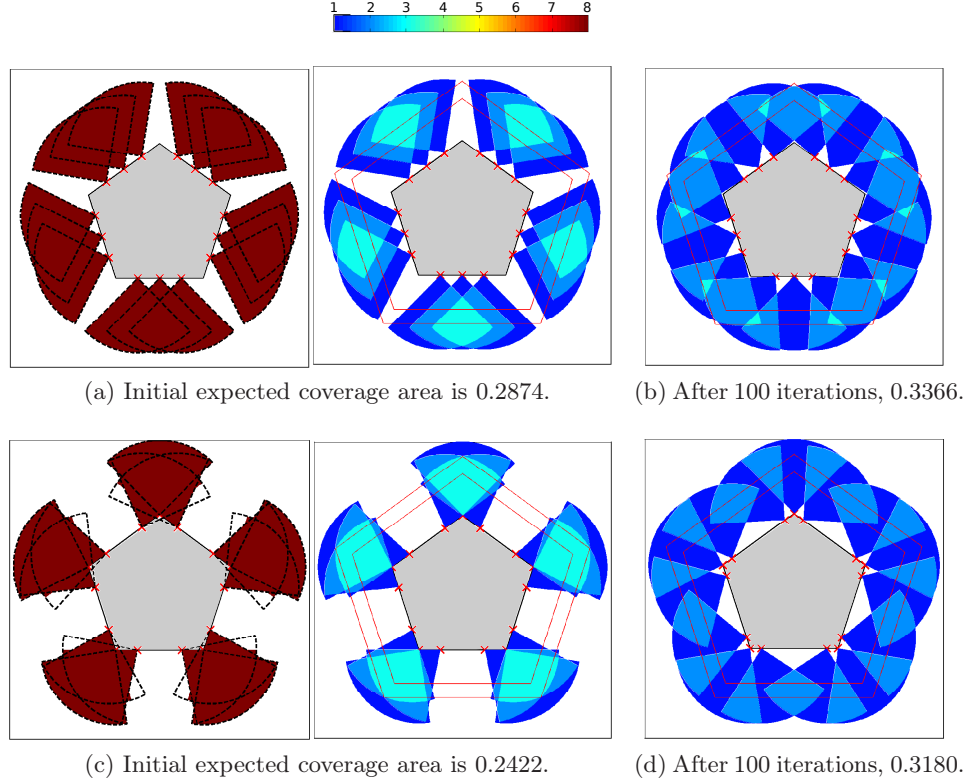


Figure 4.5: Entrance security monitoring for the pentagon under two symmetry constraints. The resulting sensor positions are obtained with 15 sensors using the same parameters $r = 0.5$, $h = 0.0025$, $\theta = \pi/2$, $p = 0.5$. There are 2 degrees of freedom in optimization.

direction normal to the boundary, and the other two move with symmetry: (1) when one sensor moves closer to the corner, the other moves to the other corner with the same distance; (2) when the viewing direction of one sensor rotates clockwise, the one of the other rotates counterclockwise at the same angle. This problem requires a 2-dimensional optimization (location and viewing direction) to position two sensors away from the center. This case, from the initial condition in Figure 4.5(a), the optimal solution is computed as Figure 4.5(b) with the expected coverage of 0.3366. Under this symmetry constraint, we find that the optimal location of three sensors divides the edge in the ratio 0.29:0.5:0.71. Interestingly, the optimal result does not equally divide the edge into four as one may expect.

As a remark, one can also consider different types of symmetry. Figures 4.5(c-d) show another of such example. Each sensor is fixed at the corner with a viewing direction exactly opposite to the center. Other two sensors move with symmetric on the edge as before. The result of our algorithm is given in Figure 4.5(d). The resulting expected value is not as high as the one in Figure 4.5(b).

We can solve the problem without any symmetry constraint. This is more suitable for the general shapes. In this case, however, the degree of freedom is increased from 2 to 30 (position and viewing direction of 15 sensors). With intermittent diffusion and gradient descent for all variables, from the initial condition in Figure 4.6(a), the optimal solution is computed as Figure 4.6(b) with the expected coverage of 0.3244. The result distributes the sensors and adjusts the viewing directions to uniformly cover the entire area of A . This result provides the better expected coverage area than the one of Figure 4.5(d) but the worse than the one of Figure 4.5(b). Although the expected coverage area is not as sharp as the one with symmetry constraint, this result shows that our algorithm can

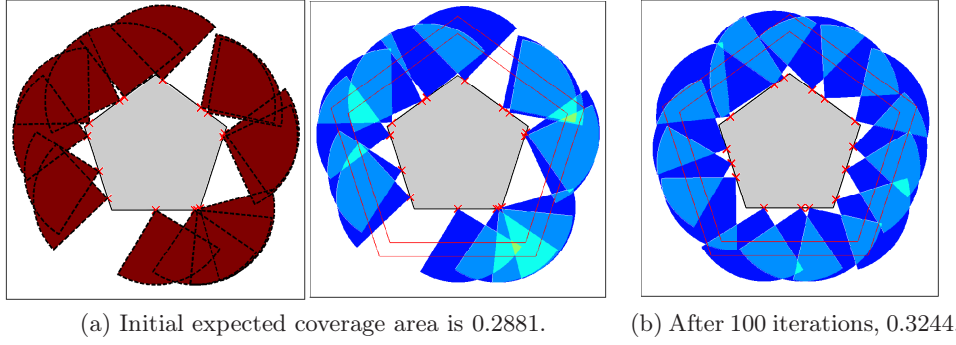


Figure 4.6: Entrance security monitoring for the pentagon. The resulting sensor positions are obtained with 15 sensors using the same parameters as Figure 4.5. There are 30 degrees of freedom in optimization.

handle various cases and approximate the globally optimal solution.

4.4 Important regions

We consider a particular problem which some regions in the monitored area are more important than others. In other words, the various weights of importance are assigned and given a priori. To apply the previous algorithm, we modify the objective functional as

$$\mathbb{E}(V(x_1, \dots, x_m, v_1, \dots, v_m)) = \int_D w(y) H(\phi(y; x_1, \dots, x_m, r_1, \dots, r_m, v_1, \dots, v_m)) (1 - p(y)) dy$$

where the weight $w(\cdot)$ is distributed over the environment.

This can take care of the scenario where there are blind spots in the environment we want to monitor. This setup resembles a realistic case, e.g., sensor positioning for grocery market and book store. To set-up this environment, we first placed a stationary long-range sensor (which can represent the location of the cashier), and computed the blind spots using the visibility computation using Algorithm 1. In Figure 4.7 (a), the circle represents the cashier's location, and the red lines behind the gray rectangles (e.g. book shelves) represents the edge of visibility boundary. In addition, near the cashier is an important area to monitor, this is represented by the red square box enclosing the red circle.

The objective is to optimally position the sensors to monitor the important regions, i.e., the two separate areas, each enclosed by the red lines. This mission can be easily tackled using our approach. We assign a weight 1 only on the important regions and 0 on the others, then apply Algorithm 3 to maximize the above functional. Considering the strategy of sensor's possible failure, the initial positions in the subfigure (b) are changed into the subfigure (c) with 50 iterations. The failure rate is chosen $p = 0.5$. As shown in the result, full coverage of the area is not achieved although the maximum of the expected coverage is attained. Instead of covering the small non-covered regions, sensors make a large overlap in the covered regions for maximum expectation because the contribution of the overlapping areas outweighs that of the non-covered areas. This is also due to the difference between the shape and the topology of the important region and the coverage shape of the sensors.

When the failure rate is lower, one can achieve full coverage. Figure 4.8 shows an example with a smaller failure rate $p = 0.1$. We apply Algorithm 3 to maximize the coverage area, and compared

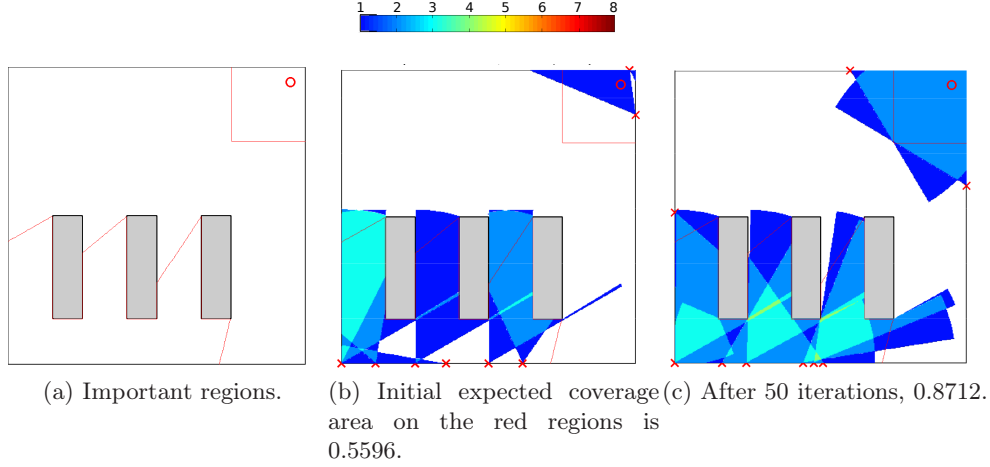


Figure 4.7: Important region monitoring with $p = 0.5$. The resulting sensor placements are obtained with 9 sensors using the parameters $r = 1.05$, $h = 0.005$, $\theta = \pi/3$. Most of the important regions are covered, but there exist non-covered regions.

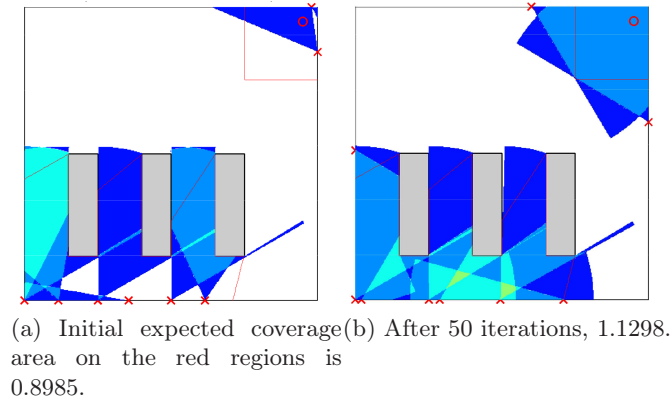


Figure 4.8: Important region monitoring with $p = 0.1$. The resulting sensor placements are obtained with 9 sensors using the same parameters as Figure 4.7. The important regions are fully covered.

to the case when $p = 0.5$, the important regions are both fully covered by the sensors, and the result gives a higher expected coverage.

4.5 Higher dimensional example

The examples presented so far were restricted to 2-dimensional environments. The proposed method can be easily generalized to the higher dimensions. By modifying the level set formulation in Section 2, the previous approach can be applicable to the higher dimensional setup. For simplicity, we focus on 3-dimensional which addresses a real world environment. The higher (> 3) dimensions can be considered in a similar manner.

A sensor in 3-dimensional space is given in Figure 4.9. The coverage of a sensor is modeled by a spherical sector whose apex point is located at the sensor location. That is, as described in the right subfigure, we generalize the 2-dimensional sensor in Figure 2.1 to the 3-dimension by rotating it about the central axis.

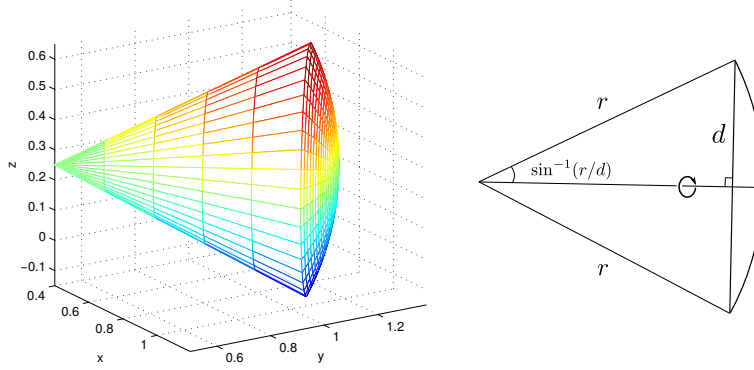


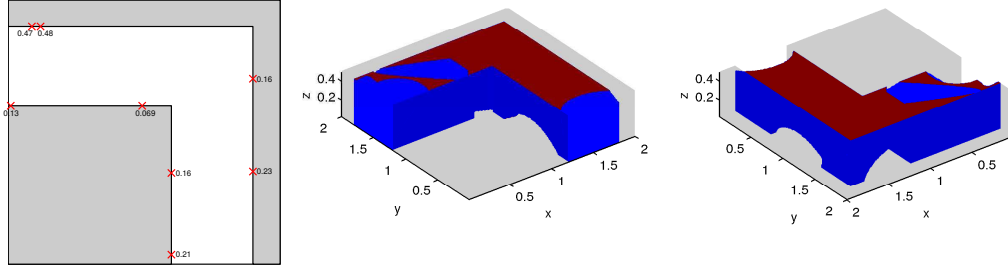
Figure 4.9: The coverage area of a 3-dimensional sensor is described by a spherical sector. The location of the sensor is at $(0.4, 0.5, 0.25)$ which is the apex point of a spherical sector. This sensor has finite sensing distance $r = 0.4$ and a base radius $d = 0.8$.

Extension to 3D can be done by following [21]. First, the level set formulation introduced in Section 2 is converted to the 3D setup. The coverage function $\phi(y; x, r, v)$ is computed by the 3D version of Algorithm 1. Then, the coverage area of multiple sensors is computed by the integration over the 3D computational domain. Secondly, we maximize the coverage area with respect to four degrees of freedom for each sensor: two for adjusting the viewing direction, and the other two for the coordinates on the boundary of the 3D obstacles. The optimization using the intermittent diffusion automatically approximates one of the globally optimal positions of the sensor in terms of four degrees of freedom.

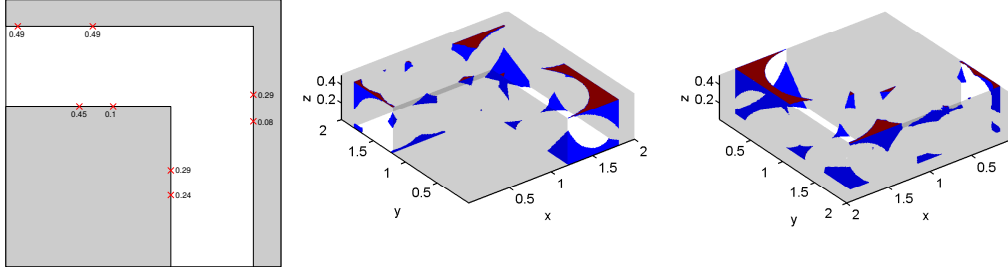
A 3D version of the example in Figure 4.3 will be constructed on a 3D computational domain whose size is $[0, 2] \times [0, 2] \times [0, 0.5]$. We place two obstacles: a square box with the dimension $[0, 1.4] \times [0, 1.4] \times [0, 0.5]$ and an upside down letter "L" shaped box with the same height. In Figure 4.10(a), the left subfigure represents an initial position using a projection onto $x - y$ plane. The numbers besides the red crosses refer to the z -coordinate of the position. The middle subfigure shows the non-monitored regions with respect to the initial position of the sensors when we look at the environment in the front. The obstacles are shaded transparent gray, the blue surface represents the boundary of uncovered regions by the sensors, and the red surface is the cut due to the top $z = 0.5$. While our algorithm changes the initial position as the subfigure (b) with 20 iterations, the expected coverage area increases from 0.1011 to 0.6259. It is also evident from the results that most of the areas are now covered under the new position of the sensors.

5 Summary

We proposed a novel method for finding the optimal position of sensors with a limited coverage range and width of viewing angle in a known environment. The efficient positioning of such equipment is increasingly important as it directly impacts the efficiency of allocated resources and system performance. We considered two global optimization problems. One is to achieve the greatest surveillance area of the region with the given number of multiple sensors which have characteristics such as range and angle limits. The other is to optimize the sensor placement given that each sensor may randomly fails to operate. In this case, the sensor placement maximizing the expected coverage area in the environment will be the desired optimal solution. From the results of several examples, we verified the effectiveness of our algorithm. The more realistic scenario where sensor coverages are correlated to each other shall be addressed in a future work.



(a) The initial expected coverage area is 0.1011. (Left) Projection of the sensor locations onto $x - y$ plane. Red crosses denote the position of the sensor and the numbers shows the height of each sensor. (Middle) A front side view in 3D. The obstacles are shaded transparent gray, the blue surface represents the boundary of uncovered regions by the sensors, and the red surface is the cut due to the top $z = 0.5$. (Right) A reverse side view.



(b) After 20 iterations, the resulting expected coverage area is 0.6259. (Left) Projection of the sensor locations onto $x - y$ plane. Red crosses denote the position of the sensor and the numbers shows the height of each sensor. (Middle) A front side view in 3D. (Right) A reverse side view.

Figure 4.10: A 3D alley with a corner. The resulting sensor positions are obtained with 8 sensors using the parameters: sensing range $r = 0.9$, base radius $d = 0.7$, step size $h = 0.01$, and failure rate $p = 0.5$. The result shows the movement of the sensors and how most of the areas are covered. Our algorithm increases the expected coverage area from 0.1011 to 0.6259.

References

- [1] A. Aggarwal. *The Art Gallery Theorem: Its Variations, Applications and Algorithmic Aspects*. PhD thesis, 1984. AAI8501615.
- [2] A. Cameron and H. Durrant-Whyte. A bayesian approach to optimal sensor placement. *The International Journal of Robotics Research*, 9(5):70–88, 1990.
- [3] L.-T. Cheng and Y.-H. Tsai. Visibility optimization using variational approaches. *Commun. Math. Sci.*, 3(3):425–451, 2005.
- [4] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE*, 89(10):1456–1477, Oct 2001.
- [5] N. A. C. Cressie. *Statistics for spatial data*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons, Inc., New York, 1991. A Wiley-Interscience Publication.

- [6] S.S. Dhillon, K. Chakrabarty, and S.S. Iyengar. Sensor placement for grid coverage under imprecise detections. In *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, volume 2, pages 1581–1587 vol.2, July 2002.
- [7] B. Engquist, A.-K. Tornberg, and R. Tsai. Discretization of Dirac delta functions in level set methods. *J. Comput. Phys.*, 207(1):28–51, 2005.
- [8] U. M. Erdem and S. Sclaroff. Optimal placement of cameras in floorplans to satisfy task requirements and cost constraints. In *In Proc. of OMNIVIS Workshop*, 2004.
- [9] G. L. Foresti, C. S. Regazzoni, and P. K. Varshney. *Multisensor Surveillance Systems: The Fusion Perspective*. Springer, 2003.
- [10] P. Frasca, F. Garin, B. Gerencsér, and J. M. Hendrickx. Optimal one-dimensional coverage by unreliable sensors. *SIAM Journal on Control and Optimization*, 53(5):3120–3140, 2015.
- [11] I. Yu. Gejadze and V. Shutyaev. On computation of the design function gradient for the sensor-location problem in variational data assimilation. *SIAM Journal on Scientific Computing*, 34(2):B127–B147, 2012.
- [12] S. H. Kang, S. J. Kim, and H. Zhou. Path optimization with limited sensing ability. *Journal of Computational Physics*, 299:887 – 901, 2015.
- [13] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, June 2008.
- [14] Y. Landa, D. Galkowski, Y. R. Huang, A. Joshi, C. Lee, K. K. Leung, G. Malla, J. Treanor, V. Voroninski, A. L. Bertozzi, and Y.-H. R. Tsai. Robotic path planning and visibility with limited sensor data. In *American Control Conference, 2007. ACC '07*, pages 5425–5430, July 2007.
- [15] Y. Landa and R. Tsai. Visibility of point clouds and exploratory path planning in unknown environments. *Commun. Math. Sci.*, 6(4):881–913, 2008.
- [16] Y. Landa, R. Tsai, and L.T. Cheng. Visibility of point clouds and mapping of unknown environments. In *Springer Notes in Computational Science and Engineering*, pages 1014–1025, 2006.
- [17] A. T. Murray, K. Kim, J. W. Davis, R. Machiraju, and R. Parent. Coverage optimization to support security monitoring. *Computers, Environment and Urban Systems*, 31(2):133 – 147, 2007.
- [18] J. O’Rourke. *Art gallery theorems and algorithms*. International Series of Monographs on Computer Science. The Clarendon Press, Oxford University Press, New York, 1987.
- [19] I. Pavlidis, V. Morellas, P. Tsiamyrtzis, and S. Harp. Urban surveillance systems: from the laboratory to the commercial world. *Proceedings of the IEEE*, 89(10):1478–1497, Oct 2001.
- [20] T.C. Shermer. Recent results in art galleries. *Proceedings of the IEEE*, 80(9):1384–1399, Sep 1992.
- [21] Y.-H. R. Tsai, L.-T. Cheng, S. Osher, P. Burchard, and G. Sapiro. Visibility and its dynamics in a PDE based implicit framework. *J. Comput. Phys.*, 199(1):260–290, 2004.

- [22] Y.-H. R. Tsai and S. Osher. Total variation and level set methods in image science. *Acta Numer.*, 14:509–573, 2005.
- [23] J. Urrutia. Art gallery and illumination problems. In *Handbook of Computational Geometry*, pages 973–1027. North-Holland, 2000.
- [24] K. Worden and A.P. Burrows. Optimal sensor placement for fault detection. *Engineering Structures*, 23(8):885 – 901, 2001.
- [25] H. Zhang and X. Wang. Optimal sensor placement. *SIAM Review*, 35(4):641–641, 1993.
- [26] H.-M. Zhou, S.-N. Chow, and T.-S. Yang. Global optimizations by intermittent diffusion. In *Chaos, CNN, Memristors and Beyond*, chapter 37, pages 466–479. Mar 2013.
- [27] Z. Zhu and M. L. Stein. Spatial sampling design for prediction with estimated parameters. *Journal of Agricultural, Biological, and Environmental Statistics*, 11(1):24–44, 2006.
- [28] D. L. Zimmerman. Optimal network design for spatial prediction, covariance parameter estimation, and empirical prediction. *Environmetrics*, 17(6):635–652, 2006.